

REMARKS/ARGUMENTS

These remarks are in response to the Final Office Action dated March 25, 2003. Claims 1-27 and 29-47 are pending in the present application. Claims 1-27 and 29-47 have been rejected. Claims 1-27 and 29-47 remain pending. For the reasons set forth more fully below, Applicant respectfully submits that the claims as presented are allowable. Consequently, reconsideration, allowance, and passage to issue are respectfully requested.

In the event, however, that the Examiner is not persuaded by Applicant's arguments, Applicant respectfully requests that the Examiner enter the arguments to clarify issues upon appeal.

Claim Rejections - 35 U.S.C. §103

The Examiner has stated:

Claims 1-27 and 29-47 are rejected under U.S.C. 103(a) as being unpatentable over Boucher et al. (US 6226680 B1). ...

As per claims 1, 2, 6, 10, 11, 15, 19, 20, and 24, Boucher teaches ... (b) adding a first header ... (see col.2, lines 35-54; col.6, lines 28-32; and col. 10, lines 18-21); ... (d) invoking a send procedure of a second and lower protocol layer of the communications protocol ... (see col.2, lines 49-54); (e) transferring ..., wherein the payload data is not copied in preparation for or during the send procedure (see col.3, lines 56-64, col.5, lines 32-37; and col.9, lines 36-52); ... (m) transferring ..., wherein the payload data is not copied in preparation for or during the receive procedure (see col.3, lines 56-64; col.5, lines 32-37; and col.9, lines 36-52).

... Although Boucher teaches ... processing "on the fly" so that all "protocol headers for a packet can be processed without copying, moving or storing the data between protocol layers" (see col. 9, lines 41-43), he does not teach that the system still processes the headers in the conventional way of adding headers to the payload data while traveling through the layers, but such step and means are well known in the art (see col. 2, lines 19-60). Therefore, it would have been obvious ... to employ the teachings of known prior ... within the system of Boucher by adding headers through the layers because Boucher teaches still teaches of processing all headers for a packet as referenced above.

As per claims 29, 38, and 39, Boucher teaches ... wherein the payload data is not copied during and between being processed by the first and second protocol layers (see col. 3, lines 56-64; col. 5, lines 32-37; and col. 9, lines 36-52).

Response to Remarks:

In response to the argument one of ordinary skill in the art would agree that the “generic protocol layer class (GPLC)” as stated in the disclosure and recited in the Amendment on page 22, is clearly taught by Boucher’s “single wider layer” (see below). Furthermore, one of ordinary skill in the art would also agree that the “common buffer”, as recited in the Amendment on page 22, is also taught by Boucher’s “buffer on a host” (see below). Furthermore, in response that the conventional method and the method of Boucher cannot be combined, it is evident in the teachings of Boucher that such methods can be combined (see Fig.4A).

In response to the argument that Boucher does not teach or suggest the “combination of processing payload data using first and second protocol layers, wherein the payload data is not copied in preparation for or during the send procedure”, clearly Boucher teaches this limitation (see col.9, line 20 to col. 11, line 42).

In response to the argument regarding Boucher, the cited reference location, by the applicant(s) or applicant’s representative, particularly column 2, lines 51-54 and column 2, lines 64-67 regarding “data moving and copying” and “repeated data copies”, is not the step of Boucher, but rather the teachings of the deficiencies of prior art and the teachings in which Boucher intends to overcome. Clearly Boucher teaches of a “protocol processing method and architecture that results effectively collapses the layers of a connection-based, layered architecture such as TCP/IP into a single wider layer which is able to send network data more directly to and from a desired location or buffer on a host” (see col. 3, lines 41-45). Furthermore, Boucher teaches “The fast-path employs a specialized microprocessor designed for processing network communication, avoiding the delays and pitfalls of conventional software layer processing, such as repeated copying and interrupts to the CPU” (see col.3, lines 56-60). Lastly, Boucher states the advantages of hardware processing rather than software processing (conventional method) is that the packets do not “have to be stored, moved, copied or pulled from storage for processing each protocol layer header, offering dramatic increases in processing efficiency and savings in processing time for each packet (see col.11, lines 43-49).

In response to the arguments regarding independent claims 2, 6, 10, 11, 15, 19, 20, 24, 29, 38, and 39, Boucher clearly teaches the element of “wherein the payload data is not copied in preparation for or during the send procedure” and/or “receive procedure” (see reference locations in claim rejection above).

Therefore, all remaining dependent claims remain rejected.

Applicant respectfully disagrees with the Examiner’s rejections. For the Examiner’s convenience, independent claim 1 is reproduced in its entirety herein below.

Claim 1

1. (previously amended) In a network having a sending computer system and a receiving computer system, each of the sending and receiving computer systems including a processor, a memory and a network adapter, the memory containing a data structure used for storing a common data buffer, a method for sending and receiving payload data by layers or sub-layers of at least one communications protocol, comprising the steps of:

- (a) storing a first start pointer pointing to a first byte of the payload data in a first common data buffer of the sending computer system;
- (b) adding a first header to the payload data in the first common data buffer at a location preceding the byte pointed to by the first start pointer according to a first protocol layer of the communications protocol at the sending computer system;
- (c) adjusting the first start pointer to point to a first byte of the first header;
- (d) invoking a send procedure of a second and lower protocol layer of the communications protocol at the sending computer system;
- (e) transferring to the second protocol layer the start pointer by the send procedure, wherein the payload data is not copied in preparation for or during the send procedure;
- (f) adding a second header to the payload data in the first common data buffer at a location preceding the first start pointer;
- (g) sending the payload data and the first and second headers to the receiving computer system;
- (h) storing the payload data and the first and second headers in a second common data buffer of the receiving computer system;
- (i) invoking a receive procedure of a second protocol layer of the communications protocol at the receiving computer system;
- (j) storing a second start pointer pointing to a first byte of the second header in the second common data buffer;
- (k) adjusting the second start pointer to point to the first byte of the first header according to the second protocol layer at the receiving computer system;
- (l) invoking a receive procedure of a first and higher protocol layer of the communications protocol at the receiving computer system; and
- (m) transferring to the first protocol layer at the receiving computer system the second start pointer, wherein the payload data is not copied in preparation for or during the receive procedure.

The present invention provides a method for allowing the sharing of code between communications protocol layers and for eliminating the need for copying payload data between

protocol layers in a sending system before the payload data is sent and between protocol layers in a receiving system after the payload data is received. A table describing seven standardized protocol layers (the “OSI Model”) can be found in the specification beginning on page 2, line 11. The method of the present invention is accomplished by the creation of a generic protocol layer class (GPLC) having send and receive procedures and a common data buffer in which sent or received payload data is placed and acted upon by each protocol layer implemented with the GPLC. Instead of copying the payload data passed to it by a higher or lower protocol layer, a protocol layer acts upon the common data buffer by moving a “start” pointer and an “end” pointer along the data contained in the common data buffer prior to invoking the next higher or lower protocol layer. Each protocol layer can then process the payload data without it having to be copied. This reduces the amount of memory and processing time required for processing the payload data (Summary). Boucher does not teach or suggest these features, as discussed below.

Boucher discloses a system for protocol processing in a computer network and includes an intelligent network interface card, which provides a fast-path that avoids conventional protocol processing for messages, greatly accelerating data communication (Abstract).

However, Boucher does not teach or suggest the *combination of* processing payload data **using first and second protocol layers**, wherein the “payload data is **not copied in preparation for or during the send procedure**” and wherein the “payload data is **not copied in preparation for or during the receive procedure**,” as recited in steps (b), (d), (e), and (m) of independent claim 1.

The Examiner has referred to column 9, line 20, to column 11, line 42, of Boucher as teaching this combination. However, this section clearly describes two separate and mutually

exclusive paths for sending messages, “a slow path 158 that uses the protocol stack of the host, or along a fast-path 159 that bypasses the protocol stack of the host” (Figure 6, and column 9, lines 36-39). The present invention as recited in independent claim 1 is directed to using protocol layers (corresponding to the protocol stack of Boucher) for sending and receiving payload data but **without copying the payload data in preparation for and during the send/receive procedures of the protocol layers**. Where Boucher discusses the protocol stack, Boucher teaches that each step typically includes “data moving and **copying**” (column 2, lines 51-54) and that “**repeated data copies** may also be necessary for the receiving host to place the data in an appropriate form at its intended destination” (column 2, lines 64-67). Applicant agrees with the Examiner that Boucher intends to overcome these deficiencies of the prior art. However, Boucher does not overcome these deficiencies by improving the use of the protocol stack, as does the present invention. Instead, Boucher provides specialized hardware that bypasses the protocol stack. Boucher teaches that a network processor “chooses between processing messages along a slow path” or “along a fast-path” (column 9, lines 36-39). Accordingly, if the slow path is chosen, messages are processed by protocol layers in the conventional manner where copying occurs, which is different from the present invention where copying does not occur. If the fast path is chosen, copying does not occur but the protocol layers are not used, which is different from the present invention where the protocol layers are used.

Furthermore, Boucher requires additional hardware such a specialized microprocessor and hardware logic (column 3, lines 56-60, column 11, lines 43-49) to process a packet, and Applicant agrees with the Examiner that Boucher states advantages of hardware processing rather than software protocol processing. However, Applicant respectfully submits that this *teaches*

away from the present invention, which achieves the sending and receiving of payload data using protocol layers (protocol processing).

Therefore, Boucher not teach or suggest the combination of steps as recited in independent claim 1, and this claim is allowable over Boucher.

Independent claims 2, 6, 10-11, 15, 19-20, and 24

Independent claims 2, 6, 10-11, 15, 19-20, and 24 recite send- and/or receive-procedures using first and second protocol layers, “wherein the payload data is not copied in preparation for or during the send procedure” and/or “receive procedure.” As described above, with respect to independent claim 1, Boucher does not teach or suggest this combination of features.

Accordingly, the above-articulated arguments related to independent claim 1 apply with equal force to claims 2, 6, 10-11, 15, 19-20, and 24. Therefore, these claims are allowable over the cited reference for at least the same reasons as claim 1.

Independent claims 29, 38, and 39

Similar to independent claim 1, independent claims 29, 38, and 39 recite steps for processing payload data using first and second protocol layers, “wherein the payload data is not copied.” As described above, with respect to independent claim 1, Boucher does not teach or suggest this combination of features. Accordingly, the above-articulated arguments related to independent claim 1 apply with equal force to claims 29, 38, and 39. Therefore, these claims are allowable over the cited reference for at least the same reasons as claim 1.

Remaining dependent claims

Dependent claims 3-5, 7-9, 12-14, 16-18, 21-23, 25-27, 30-37, and 40-47 dependent from independent claims 2, 6, 11, 15, 20, 24, 29, and 39, respectively. Accordingly, the above-articulated arguments related to claims 2, 6, 11, 15, 20, 24, 29, and 39 apply with equal force to claims 3-5, 7-9, 12-14, 16-18, 21-23, 25-27, 30-37, and 40-47, which are thus allowable over the cited references for at least the same reasons as claims 2, 6, 11, 15, 20, 24, 29, and 39.

Conclusion

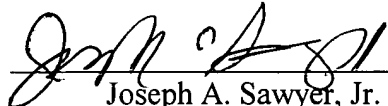
In view of the foregoing, Applicant submits that claims 1-27 and 29-47 are patentable over the cited reference. Applicant, therefore, respectfully requests reconsideration and allowance of the claims as now presented.

Applicant's attorney believes that this application is in condition for allowance. Should any unresolved issues remain, the Examiner is invited to call Applicant's attorney at the telephone number indicated below.

Respectfully submitted,

SAWYER LAW GROUP LLP

May 20, 2004
Date



Joseph A. Sawyer, Jr.
Attorney for Applicant(s)
Reg. No. 30,801
(650) 493-4540